Software Development Life Cycle at SSPL

An Summary of Methodologies We Offer

10/29/2009

Table of Contents

The SSPL Advantage

We at SSPL have management expertise in implementing several of the industry standard software development process methodologies. Some of our favorites include Waterfall, Agile, and Spiral. We are currently in the process of formulating an organizational framework of processes to conform to CMMI. This framework will embrace the best practices of various models mentioned above. So far, our choice of SDLC process methodology has always been based on what suits the customer requirement best. Some of the compelling reasons behind choosing these models in the past have been discussed in the document that follows.

Commonly Used SDLC Models at SSPL

Waterfall Model

This is one of the oldest models prevalent in the industry. It relies on sequential execution of the software development processes viz., requirements analysis, system design, development, testing/quality assurance, documentation, build and deployment, user acceptance testing, production release, and support. Despite many newer models presenting their advantages emphatically, the reasons this model is still a favorite are:

a. It offers a planned set of activities with a given beginning and an end.
b. There is no confusion for the development team as well as the clients about the activities that are to be performed, their sequence, and the end result of each activity.
c. This model insists on documenting each and every activity. Therefore, future revisions and maintenance becomes easy due to available reference. This also reduces the Knowledge Transfer costs in future.
d. The sequential activities provide reliable integration points and outputs.

Therefore, SSPL recommends use of this model in the following cases:

- When the customer has more or less clear requirements at the start,
- When there is a large number of end users generating requirements and no single point-of-contact at client end,

- When there is a greater inter-dependency of modules,
- When the concerned stakeholders are new to the concept of modular deliveries.

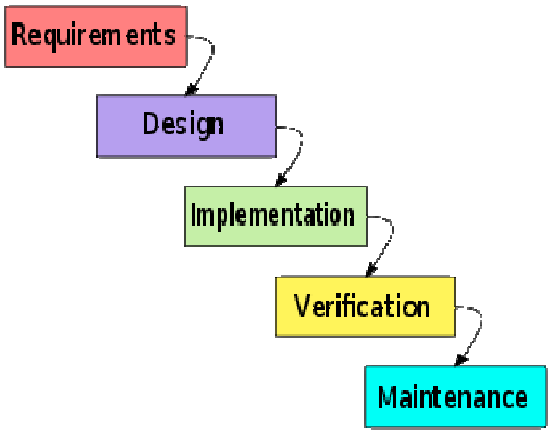

FIGURE 1: WATERFALL MODEL – SEQUENTIAL PROGRESS

Agile Model

This is one of the newest models on the block. Its greatest advantage over Waterfall model lies in the fact that it encourages frequent inspection and adaptation thereby facilitating rapid delivery of high-quality software. The reasons why this methodology finds an increasing number of followers are:

e. The frequent and small increments help minimize overall risk, and let the project adapt to changes quickly.
f. A release – albeit not marketable - is available for review at the end of each time-box thereby facilitating early feedback.
g. Frequent communication within the team promotes greater collaboration.
h. Frequent communication with the client promotes greater accountability for the team.
i. Overall timelines and costs are shrunk as compared to the conventional models.

We use many standard Agile practices such as paired programming, test-driven development, feature-driven development, code refactoring, design patterns, extreme programming (XP), deferred coding, and continuous integration, as and when we see an opportunity and suitability.

We prefer adapting Agile methodologies such as Scrum or Iterative under the following circumstances:

- When the customers do not have a clear idea about the requirements to begin with,
- When the requirements are likely to change quite frequently,
- When the users has a single point-of-contact who can act as their representative,
- When the modules or functionalities can be de-coupled sufficiently to allow fluid designs and scalable architectures.

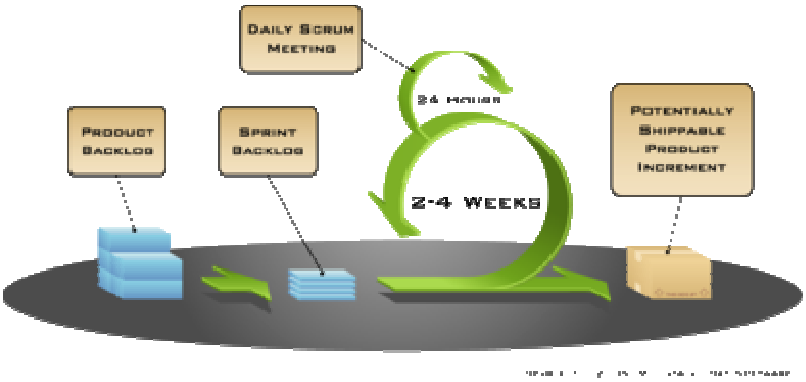

FIGURE 2: AGILE DEVELOPMENT –TIME-BOXED ITERATIVE PROGRESS

Spiral Model

This model has been around for a few decades now. Its claim to fame is that by combining the benefits of top-down and bottom-up approach by way of detailed breakdown and modularizing respectively, it helps ease the development of complex systems. Typically, this model is used for the following reasons:

j. A complete risk assessment is done at the beginning thereby enabling better risk response in later stages,
k. Iterative development of features provides functional systems at regular intervals,
l. Detailed analysis and proto-typing facilitates early identification of potential interfacing issues,
m. Modularization provides an opportunity for faster hands-on development and testing.

SSPL has had the opportunity to practice this methodology in the following cases:

- When the system is too large or complex,
- When there are constantly shifting goals or objectives of the system,
- When there are a greater number of unknowns representing threats,
- When there is sufficient time and budget allocated for analysis and design before the implementation phase.

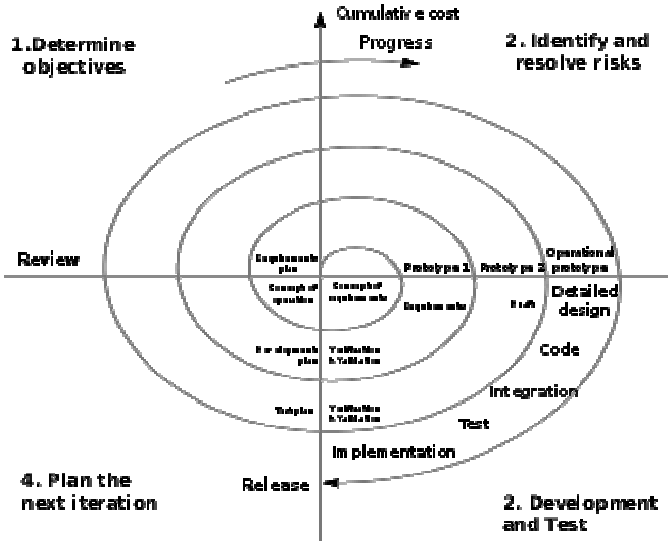

FIGURE 3: SPIRAL DEVELOPMENT – INCREMENTAL PROGRESS